



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|-------------|----------------------|-------------------------|------------------------|
| 09/925,580 | 08/09/2001 | Vugranam C. Sreedhar | YOR920010262US2 | 2877 |
| 48062 7590 09/11/2007 RYAN, MASON & LEWIS, LLP 1300 POST ROAD SUITE 205 FAIRFIELD, CT 06824 | | | EXAMINER VO, TED T | |
| | | | ART UNIT 2191 | PAPER NUMBER |
| | | | MAIL DATE 09/11/2007 | DELIVERY MODE PAPER |

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

MAILED

SEP 11 2007

Technology Center 2100

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 09/925,580

Filing Date: August 09, 2001

Appellant(s): SREEDHAR, VUGRANAM C.

Kevin M. Mason
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 06/04/2007 appealing from the Office action mailed 11/21/2006.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is substantially correct. The changes are as follows:

Claims 2-4, 10-12, 18 are rejected under 35 U.S.C. 102(b) as being anticipated by Posnak, Lavender et al "Adaptive Configuration an Object Structural Pattern for Adaptive Applications", 1996.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

Posnak, et al, "Adaptive Configuration an Object Structural Pattern for Adaptive Applications", 3rd Pattern Languages of Programming Conference, Monticello, Illinois, (September 1996), pp. 1-10.

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claims 2-4, 10-12, 18 are rejected under 35 U.S.C. 102(b) as being anticipated by Posnak, Lavender et al "Adaptive Configuration an Object Structural Pattern for Adaptive Applications", 1996 (Hereinafter: Lavender).

It should be noted that:

Claim 4 is an independent claim. Claims 2-3 depend on Claim 4.

Claim 12 is an independent claim. Claims 10-11 depend on Claim 12.

Claim 18 is an independent claim.

Appellants stated that they only appeal on Claims 4, 12, and 18 and they do not appeal on Claims 2-3 that depend on Claim 4, and Claims 10-11 that depend on Claim 12.

Accordingly, all dependent claims will stand all fall together with Claims 4, 12, and 18.

It appears that the argument is solely based only on the rejection of Claim 4. It appears that Appellants group Claims 4, 12, and 18 as the same group.

As per Claim 4: Lavender discloses,

A method executed by a processor for programming a software component, said method comprising the steps of:

defining properties of said software component (e.g. components as shown in Figure 1 (p. 2), where a component (see p.3) can be defined as a consumer, producer etc. E.g., a consumer component defines the interface for inputting data or control information to a protocol component. This also defines the component's type for the purposes of composition), ***including at least one input port and at least one output port*** (See Figure 3, input/output);

providing a software mechanism for instantiating said software component (See p. 1: sec. 2: Motivation, see second paragraph: ‘..various algorithms may be instantiated...’. Note: the reference shows consumer, producer which are classes of object-oriented software components that are instantiated at run-time); ***and***

utilizing an attach command (i.e. the instruction “bool attached” in the class Port given in p. 6, or instruction “Attach()” in the reference) ***to attach at least one of said at least one input port to a class*** (See p. 8: show various ways of using “attach()” to attach one element to another element. See three explicit classes: Class Consumer, Class Producer, Class Port, particularly refer to connection between Port Class and Consumer class. In p. 6, the reference discusses the use of Port class that encapsulates the attachment (p. 6: Management and use of this set is simplified by the use of a Port class, which encapsulates the state of an attachment to another module). In p. 4, The port object passes the data to M2 by invoking M2's Input() method. After

Art Unit: 2191

M2 has transformed its input data, it can then forward the data on to the next module in the same fashion, then see the program instructions/statements in the Port class shown in p. 6:

```
Consumer<O> *consumer;    //the module attached to this port
Int inputPort;            // input port attached to this port
bool attached;            // true if a module is attached
```

The teaching shows: An input port is attached to the port of a port class. The attachment shows in the same manner of the claim. As seen in the reference, the attachment is performed by “attach()” or the instruction “bool attached”.

As per Claims 12, and 18: These claims are rejected in the same manner as reasoned in the rejection of claim 4.

With regards to claims 2-3, 10-11: These claims are not under appeal.

As per claim 2: Lavender further discloses,

further comprising the step of allowing said software component to access an external environment only through said output port (See page Figure 3, connection component to component, using the output port).

As per Claim 3: From further limitation of Claim 1, Magee further discloses,

further comprising the step of allowing a client to access said software component only through said input port (See Figure 1, See Media Object, graphically, it shows such accessibility).

As per claims 10-11: Claims are rejected in the same reasons as set forth in connecting to the rejections of Claims 2, and 3, because the claimed functionality is corresponding to the functionality of Claims 2 and 3, respectively. See rationale as discussed above for Claims 2-3.

(10) Response to Argument

Appellants' issue on appeal is relied only on the limitation in the claim 4:

“utilizing an attach command to attach at least one of said at least one input port to a class”.

In the argument, Appellants depicted a sequence of instructions in a program in their specification:

```
attach xin BoolClass; // attach input port xin to class".  
BoolClass;  
....  
....
```

Appellants mapped this sequence of instruction for supporting the limitation of the claim 4. Appellants alleged that Lavender does not “utilize an attach command to attach an input port to a class”. Appellants noted that the “attach” in Lavender attaches an output port of one module to an input port of another module (Brief: p.4).

Art Unit: 2191

Response to Appellants' argument:

Examiner disagrees with Appellants' argument that Lavender fails to disclose "*utilizing an attach command to attach at least one of said at least one input port to a class*".

It should be noted that Lavender, whose teaching is also object-oriented subject, shows the instruction "attach()", or a sequence of instructions addressed in the Port class shown in p. 6:

```
Int inputPort;    // input port attached to this port  
bool attached;   // true if a module is attached
```

The instruction attach() used in the reference of Lavender or the programming instruction "bool attached" as in the program in p. 6 is a syntactical instruction which is the same as the "attach command" as in the Appellants' program. Its functionality is clearly binding one programming element to another programming element as the attachment in the program of the Appellants.

It should be noted that, the Appellants' claims merely recite: "*utilizing an attach command to attach at least one of said at least one input port to a class*".

It should be noted that Command is an instruction to a computer program that, when issued by user, causes an action to be carried out.

The "attach command", according to the Appellants' specification, is merely a syntactical instruction in a sequence of instructions in a program. This instruction is syntactical with "attach", and associated with some parameters:

(Appellants)

```
attach xin BoolClass; // attach input port"  
BoolClass;
```


Art Unit: 2191

It can be seen in the Lavender prior art, Lavender clearly discloses the attach command:

(Lavender)

```
Int inputPort;    // input port attached to this port
bool attached;    // true if a module is attached
```

Lavender teaches connecting an input port to a port class. According to the program in p. 6, the “input port” is from a module and is attached to the port class. See in the p. 2, within the paragraph starts with “A concrete example.... reconfigurability requirements.”, at line 6, Lavender wrote,

“The modules in the pipeline are shown as having different shapes to illustrate the fact that all modules use a uniform input/output mechanism, which is parameterized by different input and output data types. The module interfaces, as defined by these data types, must match in order to obtain a syntactically correct, type-safe configuration”.

Thus, the component’ structure of Lavender includes the modules using uniform input/output mechanism. The Port class is used as the interface class, as among the claimed limitation: “*to a class*”, where the attachment is performed by instruction “attached ()” as discussed throughout the reference of Lavender, or as shown in the program in p. 6:

```
Int inputPort;    // input port attached to this port
bool attached;    // true if a module is attached
```

Therefore, Lavender clearly discloses “*utilizing an attach command to attach at least one of said at least one input port to a class*”. The rejection should be sustained.

Art Unit: 2191

(11) Related Proceeding(s) Appendix

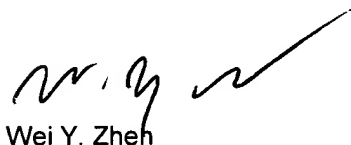
No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

TTV

Conferees:



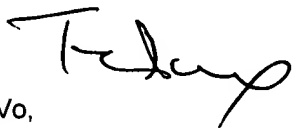
Wei Y. Zhen

SPE of Art unit 2191

Eddie C. Lee



EDDIE C. LEE
SUPERVISORY PATENT EXAMINER



Ted Vo,

Primary Examiner